<https://reacttraining.cn/web/example/basic>

见素材react-router

一、**基**本使用

import {

BrowserRouter as Router,

Link，

Route,

} from 'react-router-dom'

<Router>

<div>

<Link to="/">A</Link>

<Link to="/b">B</Link>

<Route path="/" exact component={A}/>

<Route path="/b" component={B}/>

</div>

</Router>

const B = ({ match }) => (

<div>

<Link to={`${match.url}/c`}>C</Link>

<Route path={match.url} exact render={() => (

<div>正好匹配</div>

)}/>

<Route path={`${match.url}/:cUrl`} component={C}/>

</div>

)

const C = ({ match }) => (

<div>

<h3>{match.params.cUrl}</h3>

</div>

)

二、URL参数

三、**认**证

import {

BrowserRouter as Router,

Link，

Route,

withRouter，

Redirect,

} from 'react-router-dom'

const 组件 = withRouter(({ history }) => (

<div>

//可以使用history.push('/')等

</div>

))

<Redirect to={} />

<Redirect from="/路由" to="/路由" />

四、**自**定义链接，以下暂时不背

<MyLink myTo="/" myExact={true} label="首页" />

const MyLink = ({ myTo, myExact , label }) => (

<Route path={myTo} exact={myExact} children={({ match }) => (

<div>

<Link to={myTo}>{label}</Link>

</div>

)}/>

)

五、**阻**止导航

import {

BrowserRouter as Router,

Link，

Route,

Prompt

} from 'react-router-dom'

<Prompt

when={变量值为true或false}

message={location => (

`${location.pathname}`

)}

/>

六、**未**匹配

<Route component={NoMatch}/>

const NoMatch = ({ location }) => (

<div>

无法匹配{location.pathname}

</div>

)

七、路径递归

八、侧边栏

九、过渡动画

十、**模**糊匹配

import {

BrowserRouter as Router,

Link，

Route,

Switch

} from 'react-router-dom'

<Switch>

<Route />

</Switch>

Switch内的第一个匹配路由会被渲染出来。

十一、路由配置

十二、模态画廊

基认自阻未模